

Construyendo FrankLab: Una Plataforma Web de Robótica Educativa

Juan de la Fuente, Marcos Picucci, Daiana Bonet Peinado,
Rafael Zurita, Gerardo Parra, Jorge Rodríguez, and Laura Cecchi

Grupo de Investigación en Lenguajes e Inteligencia Artificial
Departamento de Teoría de la Computación - Facultad de Informática
Universidad Nacional del Comahue
Buenos Aires 1400, Neuquén, Argentina
{juan.delafuente,marcos.picucci,daiana.bonet,
rafa,gparra,j.rodrig,lcecchi}@fi.uncoma.edu.ar

Resumen El movimiento *CS for All* ha logrado captar la atención durante los últimos años. En este contexto se evidencia la necesidad de desarrollar recursos didácticos específicos al campo de la Educación en Ciencias de la Computación que posibiliten una mejor aproximación de la disciplina al ámbito de la educación obligatoria.

En este trabajo se presenta una plataforma Web de robótica educativa, *FrankLab*, diseñada para favorecer el desarrollo de los procesos de enseñanza y de aprendizaje de conceptos y prácticas en el área Algoritmos y Lenguajes de Programación durante toda la educación obligatoria.

FrankLab es un entorno de programación en línea que permite a estudiantes de escuelas primarias y secundarias programar robots educativos de la familia Frankestito. Dicho ambiente está compuesto por un conjunto de herramientas de programación que soporta Programación Tangible, Programación Basada en Bloques y Programación basada en texto, como Python. Los estudiantes ejecutan sus programas desde un navegador web sobre robots físicos o sobre un ambiente simulado.

Keywords: Robótica Educativa, Educación en Ciencias de la Computación, Enseñanza de la Programación, Entorno Web para Robótica Educativa, Programación Tangible, Programación Basada en Bloques

1. Introducción

El movimiento *CS for all*[9] ha recibido considerable atención durante los últimos años. En este marco se construye acuerdo en relación a la necesidad de ampliar la participación de las Ciencias de la Computación en las propuestas de enseñanza para la educación obligatoria, como forma de aumentar las posibilidades de comprensión del mundo y como política para contribuir a democratizar el acceso, para todos los estudiantes, al conocimiento producido en el ámbito de la computación [3,9].

2 Construyendo FrankLab: Una Plataforma Web de Robótica Educativa

La Robótica Educativa se presenta como un dispositivo pedagógico utilizado con frecuencia, en diferentes niveles del sistema educativo, para enseñar Ciencias de la Computación a estudiantes de todas las edades.

En este contexto existe un amplio consenso sobre que las experiencias educativas que involucran la programación de robots tienen un gran potencial para mejorar los procesos de enseñanza de aspectos de la computación, que frecuentemente se presentan como difíciles de percibir y comprender.

Una de las principales potencialidades de la Robótica Educativa es exponer y hacer tangibles conceptos abstractos de la computación. De esta forma, se favorece la construcción de conocimiento principalmente, en el área de conocimiento Algoritmos y Lenguajes de Programación. Además, se promueve el desarrollo de habilidades enmarcadas en el pensamiento computacional, como la resolución de problemas y el trabajo colaborativo.

Los trabajos realizados en el ámbito del desarrollo de plataformas de robótica educativa son numerosos y demuestran efectividad para facilitar la comprensión de conceptos fundamentales de la computación, que generalmente presentan dificultad en su enseñanza. Sin embargo, persisten campos problemáticos no explorados suficientemente en la dirección de ubicar la premisa *CS for all* y, en particular, las experiencias con robots en el plano de lo posible para más niños y adolescentes: en general las plataformas no soportan múltiples modos de programación, lo que reduce el grupo de estudiantes al que se orienta la propuesta. El acceso a los robots físicos suele ser escaso, por lo que las actividades que involucren su programación se desarrollan en espacios y tiempos limitados. Finalmente, pocas plataformas soportan capacidades que posibiliten la colaboración masiva.

Teniendo en cuenta este contexto, en este trabajo se presenta FrankLab, una plataforma Web de robótica educativa diseñada con el fin de mejorar las posibilidades de aprender Ciencias de la Computación, ampliando la población estudiantil destinataria. Así, estudiantes desde el nivel inicial hasta los últimos años de la educación secundaria, e inclusive aquellos que hayan superado el nivel medio, o aquellos que tengan alguna discapacidad dispondrán de herramientas de programación adecuadas a sus particularidades.

Esta plataforma, basada en tecnologías Web, soporta múltiples modos de programación en línea: tangible, basada en bloques, basada en texto y posibilita el abordaje de problemas complejos. Asimismo, busca mejorar la disponibilidad de los recursos, posibilitando el acceso remoto a laboratorios físicos y simulados. En cuanto a la gestión de programas, integra características como compartir, comentar y remixar proyectos para promover la colaboración masiva.

FrankLab tendrá licencia open source o compatible de acuerdo a las tecnologías Web utilizadas en su desarrollo y será accesible en forma pública y gratuita.

Este trabajo se organiza de la siguiente manera: en la siguiente sección se describe el marco de referencia para el desarrollo de FrankLab. En la sección 3 se presenta la plataforma Web de robótica educativa, analizando su estructura, los robots físicos a los que está orientada y los modos de programación que soporta. Por último, en la sección 4 se detallan las conclusiones.

2. Marco conceptual

Enseñar y que los estudiantes aprendan a programar presenta diferentes dificultades y posibilidades de acuerdo a la edad y singularidad del aprendiz. Además del desafío de diseñar soluciones algorítmicas para resolver problemas específicos, el estudiante debe desarrollar competencias para expresarlas en un lenguaje en particular.

El pensamiento computacional es una habilidad fundamental para todos, no solo para los informáticos. Entre muchos otros aspectos, pensamiento computacional es usar abstracciones y descomposiciones al intentar solucionar una tarea compleja o al diseñar un sistema de tamaño considerable. Es elegir una representación apropiada de un problema o modelar los aspectos relevantes del problema para hacerlo tratable. Normalmente, también requiere la capacidad de manejar múltiples niveles de abstracción.

La programación tangible ofrece oportunidades de aprendizaje a niños y adolescentes no alfabetizados. En este modo de programación, una colección de objetos físicos se convierten en elementos de un lenguaje reduciendo significativamente la edad para participar de este tipo de actividades formativas [1,7,8].

Por otro lado, el enfoque de programación basado en bloques ayuda a definir fácilmente el concepto de pensamiento computacional basándose en tres dimensiones: concepto, práctica y perspectiva computacional en un lenguaje de programación, que cumple con la difícil tarea de tener “piso bajo”, “techo alto” y “paredes anchas”. Esto es, facilidad para comenzar a programar, oportunidades de crear proyectos de complejidad creciente y soportar diversidad de tipos de proyectos e intereses, características primordiales para aprender a programar [2].

En trabajos anteriores a este artículo, presentamos la arquitectura de Frankestito, un robot Myro compatible de bajo costo, diseñado y construido bajo licencias libres, para ser utilizado como recurso didáctico disciplinar. Una característica distintiva de este robot educativo es la incorporación de un sistema de visión y comunicación Wi-Fi [4]. Teniendo en mente las ventajas de las capacidades mencionadas, se trabajó sobre el robot educativo Multiplo N6 Max, modificando su arquitectura de hardware y de software embebido, a fin de hacerlo compatible con Frankestito[10].

El robot educativo Frankestito se utiliza regularmente para enseñar Algoritmos y Lenguajes de Programación a estudiantes secundarios, en actividades de divulgación científica e investigación [6].

Los trabajos descriptos en esta sección presentan abordajes parciales al campo problemático en cuestión. Se considera que el tratamiento en estas áreas debe plantearse como integrado y estudiarse en forma articulada para avanzar en la elaboración de una plataforma que logre recuperar aportes realizados, en función de ampliar la posibilidades de concretar experiencias educativas basadas en la robótica educativa.

En este sentido, FrankLab es una solución integradora que intenta recuperar las potencialidades descritas en esta sección.

4 Construyendo FrankLab: Una Plataforma Web de Robótica Educativa

3. FrankLab

FrankLab es una herramienta Web diseñada con la finalidad de proveer un ambiente para la programación de robots educativos compatibles con Frankes-tito. Este entorno permite la codificación en línea de programas, los cuales son luego ejecutados por un robot físico o simulado, cuyo accionar puede ser visualizado en tiempo real por los usuarios.

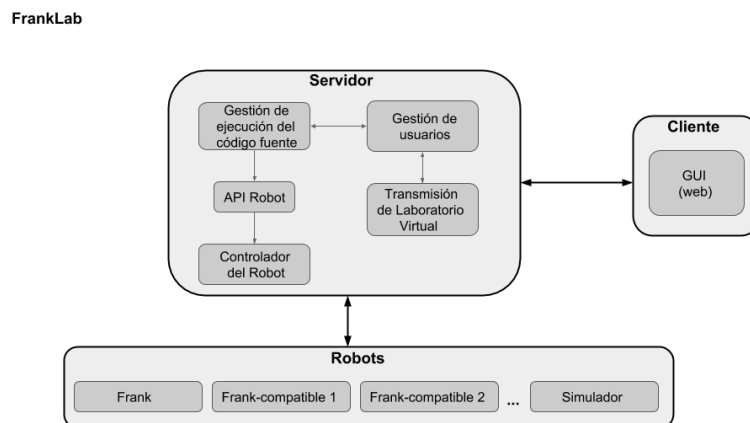


Figura 1. Estructura FrankLab

3.1. Estructura de la Plataforma

La plataforma, que se muestra en la Figura 1, está basada en el modelo de comunicación cliente-servidor. La estructura lógica del servidor está agrupada en cinco módulos, distribuidos en base a su funcionalidad:

- *Gestión de usuarios*: Este módulo aborda, por un lado, problemas en relación al modelo de datos: roles y privilegios de acceso de los distintos tipos de usuarios (estudiante, profesor y administrador) y administración de perfiles y archivos locales a cada uno de éstos. Por otro lado, comprende aspectos en relación a la edición de código fuente y a la colaboración que puede existir entre un grupo de alumnos sobre el mismo programa. Asimismo, cubre aspectos sobre la administración de los turnos que dispone cada usuario, con el objetivo de lograr un uso equitativo de los recursos físicos.
- *Transmisión de Laboratorio Virtual*: Este módulo es el encargado del proceso de entrega de los datos de la visualización en vivo, capturada por la cámara del laboratorio remoto hacia el navegador del cliente.

- *Gestión de ejecución del código fuente:* Este módulo es el encargado de realizar la compilación o interpretación del código fuente, para los lenguajes que así lo requieran.
- *API del robot:* Un lenguaje común para todos los robots físicos o simulados es definido y utilizado para controlarlos, de forma de proveer una capa de abstracción entre el lenguaje de programación elegido y la implementación real. Este módulo es el encargado de la traducción del comando de alto nivel al conjunto de instrucciones de bajo nivel que lo implementan.
- *Controlador del robot:* Este módulo gestiona las rutinas de bajo nivel que dan soporte al manejo y comunicación de robots físicos o simulados. Ejemplos de ellas son, el análisis de las imágenes capturadas por la cámara del robot y la transmisión y recepción de datos hacia y desde el mismo, respectivamente.

En lo que respecta a la interacción del cliente con el servidor, inicialmente el usuario debe acceder mediante su navegador a la página principal de FrankLab e iniciar una sesión identificándose.

Una vez validada su identidad, tendrá acceso a la selección del modo de programación a utilizar, esto es: tangible, basada en bloques o basada en texto. Dependiendo de su elección, se despliegan los lenguajes soportados para el modo elegido; por ejemplo, Python y C++ en el caso de la programación textual, e inicialmente Blockly cuando se trata del enfoque de programación por bloques.

Concretada la selección, se provee al usuario del respectivo entorno de codificación para la edición de sus programas. Al ser éstos ejecutados, el usuario puede visualizar en directo lo que la cámara principal del robot y del laboratorio remoto están capturando en ese momento, es decir, la vista en primera y tercera persona de los movimientos del robot, basados en la ejecución de su programa.

En el caso de no haber disponibilidad de robots para su utilización, se puede en forma alternativa visualizar la ejecución del programa en un ambiente simulado. Por último, dado que el usuario puede proveer y recibir feedback hacia y desde el robot, se dispone de una sección para ingresar/recibir datos de entrada/salida e informar acerca del estado actual de la ejecución.

3.2. Robots Físicos

Frankestito es una familia de robots físicos diseñados y contruidos como hardware y software libre en la Facultad de Informática. Actualmente, el modelo más utilizado en las actividades de programación con robots es el N6-Frankestito[4,6,10].

Frankestito y N6-Frankestito son robots educativos con visión. Físicamente están compuestos de un chasis con dos motores continuos y una tercera rueda de movimientos libres. La potencia de los motores está gestionada por un microcontrolador más un integrado para tal fin. También, sobre el chasis, se encuentra empotrado una computadora de una única placa (del inglés SBC: Single Board Computer) con un microprocesador MIPS, interfaz usb (en donde está conectada una cámara de video), interfaz Wi-Fi (para comunicaciones TCP/IP)

6 Construyendo FrankLab: Una Plataforma Web de Robótica Educativa

y un dispositivo UART para la comunicación serie entre los dos procesadores (microcontrolador y microprocesador MIPS).

El software está compuesto de un firmware que se ejecuta en el microcontrolador, y una versión de Linux embebida en la SBC. El firmware interpreta el nivel mas bajo del protocolo de MyRO-Scribbler , y controla los motores en base a los paquetes enviados y recibidos mediante la interfaz UART. En el Linux embebido existe, por otro lado, un conjunto de programas que se encarga de la captura de video de la cámara, y de la recepción de comandos y envío de información a los programas clientes. De esta manera, un programa cliente puede, a través de una comunicación Wi-Fi, controlar los motores del robot, y sensar del ambiente mediante mensajes sencillos definidos en el protocolo.

3.3. Modos de Programación Soportados

Con la intención de hacer de la programación de robots una experiencia posible para más estudiantes, FrankLab soporta diferentes modos de programación que la hacen accesible a niños y adolescentes desde el nivel inicial hasta después de la Escuela Secundaria. La Figura 2 muestra los modos de programación soportados por FrankLab en relación a los niveles del sistema educativos.

La categoría Modos de Programación como forma de organización teórico conceptual y su relación con diferentes momentos de la escolarización obligatoria corresponden a construcciones teóricas elaboradas por los autores de este documento.

Nivel en el Sistema educativo	Modos de Programación Soportados		
	Tangible	Basado en Bloques	Basado en Texto
Nivel Inicial			
Primaria Primeros años	FrankTangible		
Primaria Últimos años		Blockly	
Secundaria Primeros años			
Secundaria Últimos años			Python Java C++
Más allá del secundario			
	FrankLab = Laboratorio Físico + Laboratorio Simulado + Entorno Web + Colaboración Masiva		

Figura 2. Modos de Programación soportados por FrankLab

Programación Tangible La Programación Tangible que incorpora el entorno *FrankLab*, permite programar robots educativos compatibles con *Frankestito* mediante la manipulación directa de objetos físicos, como estrategia para reducir el

esfuerzo cognitivo y la habilidades previas requeridas para aprender conceptos fundamentales sobre Algoritmos y Programación.

Este enfoque permite que Frankestito llegue, en los procesos de enseñanza de la programación, a una población más amplia de estudiantes, incluyendo aquellos no alfabetizados. Es necesario destacar que, al trabajar con objetos físicos nos brinda la posibilidad de *integración* en las clases de programación, es decir, unificar la herramienta de trabajo para todos los estudiantes sin distinción.

En este ambiente se trabaja con una colección de piezas tangibles, como bloques de madera o trozos de cartulina, que se secuencian para construir programas. Cada pieza corresponde a un comando o estructura de control, de forma similar a lenguajes basados en bloque.

A diferencia de otros entornos donde se trabaja con programación física, estas piezas no utilizan ningún tipo de hardware, permitiendo que el factor económico no sea un impedimento para la enseñanza de Ciencias de la Computación.

De este modo, se trabaja inicialmente con los objetos hasta lograr la secuencia buscada, luego se requiere tomar una fotografía al trabajo realizado para ser enviado al robot a través de FrankLab y poder observar los resultados del programa creado.

Para que esto sea posible, es necesario definir un lenguaje de programación tangible que permita representar las principales funcionalidades de Frankestito y las estructuras de control básicas. Además, se considera el desarrollo de un *parser* que traduzca especificaciones en este lenguaje a uno de los soportados por los robots educativos compatibles con Frankestito.

En este sentido, en la plataforma FrankLab se desarrolla un apartado que incluya el lenguaje con objetos físicos, el parser para producir un código ejecutable a partir de esos objetos, el medio en que esos programas se ejecutan, en nuestro caso los robots físicos.

Teniendo en cuenta la estructura planteada, en la Figura 1, de FrankLab, los siguientes módulos ejecutarán acciones específicas respecto a este modo de programación:

- **Cliente:** Luego de que los estudiantes definen la secuencia de figuras, es necesario transferir el código generado al robot que lo ejecute. Para ello se toma una fotografía del programa construido con objetos físicos, que es subida a la plataforma FrankLab en la sección de *programación tangible*.
- **Gestión de Ejecución del Código Fuente:** En este módulo la imagen es procesada para identificar las distintas funcionalidades y estructuras utilizadas en el programa. Luego mediante el *parser* se traducirá del lenguaje con objetos físicos a un lenguaje común para todos los robots compatibles con Frankestito.

Programación Basada en Bloques La programación basada en bloques, referencia a la posibilidad de generar código comprensible por *Frankestito* mediante una gramática basada en “bloques de programación”, gráficos apilables e interactivos. Los mismos permiten generar programas sin recurrir a un lenguaje

8 Construyendo FrankLab: Una Plataforma Web de Robótica Educativa

de programación basado en texto, lo que reduce la curva de aprendizaje en la utilización de los robots educativos.

La idea es dotar a *FrankLab*, con bloques de programación que representen todas las funciones con las que cuenta *Frankestito*, a fin de ser utilizados dentro del entorno para controlar el robot. De este modo, se logra un entorno de programación sencillo y transparente con comunicación efectiva y simplificada.

Cada bloque tiene una estructura diferenciada por tipo de instrucción, permitiendo que al apilarlos produzca porciones de código sintácticamente correctas, facilitando el aprendizaje de las estructuras de programación de forma gráfica y explícita.

Además, estos bloques concentran toda la funcionalidad provista por la arquitectura de los robots, por lo que la expresividad del lenguaje representado en los bloques es análoga a cualquier otro lenguaje de la plataforma, permitiendo que los proyectos puedan ser interpretados o traducidos si el usuario cambia de modo o lenguaje de programación.

En la Figura 1 se puede observar como esta funcionalidad se encuentra distribuida entre dos módulos bien diferenciados:

- **Cliente:** Del lado del *Cliente*, se encuentra la ejecución de las librerías que proveen los bloques gráficos al navegador Web, permitiendo la utilización de los mismos en el ambiente de programación.
- **Gestión de Ejecución del Código Fuente:** una vez que el usuario decide *ejecutar* el código desde su navegador, el mismo es traducido a un código equivalente en un lenguaje de alto nivel basado en texto. Asimismo, este módulo es quién interactúa con el código traducido desde los bloques hacia el lenguaje intermedio del robot.

Es necesario aclarar la separación entre los bloques gráficos de programación y el lenguaje representado por los mismos. Los *bloques gráficos* son una implementación propia en base a las librerías del proyecto de código libre de programación visual *Blockly*, que provee una gramática de componente visual para facilitar la codificación de diversos lenguajes de programación[5]. En la Figura 3 se muestra un ejemplo de bloques de programación para *Frankestito* desarrollada en el ambiente *Blockly*.



Figura 3. Ejemplo de bloques de programación para *Frankestito*

Por otro lado, el lenguaje que representan y que, en última instancia, es generado por el módulo de *Gestión de Ejecución del Código Fuente* de la plataforma es propio de la arquitectura *Frankestito* y corresponde propiamente al robot[4].

Programación basada en texto La plataforma FrankLab ofrece a los estudiantes la posibilidad de seleccionar el lenguaje de programación basado en texto de su preferencia, entre los soportados por Frankestito, para codificar sus programas. Es posible interactuar con el robot a través de una diversidad de lenguajes de programación como Python, C y C++, entre otros.

La implementación incorpora un entorno de desarrollo integrado en línea, que consiste en un editor de textos para programar, al que se accede a través de un navegador Web. Esta opción permite editar código fuente sin la necesidad de instalar un IDE en la máquina del usuario, reduce los tiempos de instalación y minimiza los requerimientos de hardware. Por otro lado, permite programar desde cualquier lugar y dispositivo que cuente con un navegador Web instalado.

- **Cliente:** el editor de código en línea permite a los estudiantes seleccionar el lenguaje de programación y codificar sus programas. Integra las características habituales de un entorno de desarrollo integrado en línea, para facilitar el proceso de codificación, como por ejemplo por ejemplo, coloreado de palabras reservadas, reconocimiento de sintaxis y autocompletado de código. En conjunto estas funcionalidades contribuyen a reducir la curva de aprendizaje.
- **Gestión de Ejecución del Código Fuente:** En este módulo se gestiona la compilación o interpretación del código. Asimismo, se espera que en casos de detectar errores del tipo léxico o sintáctico, se provea asistencia al usuario mediante la descripción del error encontrado.

La Plataforma Web de Robótica Educativa presentada en este trabajo permite ir más allá de la resolución de problemas clásicos con el objeto de enseñar a programar. El *techo* dado por la enseñanza de la programación abordando problemas simples, sino que es posible atacar problemas de mayor complejidad considerados en la categoría de la investigación o la innovación. Ejemplos de problemas en esta clase son, entre otros:

- Procesamiento de imagen: actualmente los robots compatibles con Frankestito pueden detectar colores, su cantidad y posición. Una mejora podría ser la detección de figuras geométricas o de rostros en una imagen.
- SLAM (Simultaneous Localization And Mapping): los robots navegan en ambientes desconocidos. La construcción y actualización de un mapa del sector, mientras que realiza una estimación de su ubicación y trayectoria, permitiría ampliar el conjunto de aplicaciones para los robots compatibles con Frankestito como vehículos autónomos.
- Sistemas Multiagentes: las aplicaciones que involucran un conjunto de agentes coordinados para lograr un objetivo común, requieren de comunicación sincronizada entre ellos. Ejemplo de esto, es la búsqueda de sobrevivientes en una catástrofe por grupos de robots autónomos.

4. Conclusiones

En este trabajo se ha presentado FrankLab, una plataforma Web de robótica educativa delineada como una solución integradora a características detec-

10 Construyendo FrankLab: Una Plataforma Web de Robótica Educativa

tadas en la enseñanza de Ciencias de la Computación, como por ejemplo, la singularidad del aprendiz, los lenguajes de programación adecuados para lograr competencias particulares y la escasa disponibilidad de los recursos físicos, entre otras.

Su arquitectura basada en el modelo cliente-servidor y estructurada en forma modular, posibilita el soporte de diferentes modos de programación (tangible, por bloques y basada en texto) y de diversos lenguajes de programación. Así, de acuerdo a su diseño, la plataforma que fue proyectada con el fin de mejorar las posibilidades de aprender Ciencias de la Computación, permite ampliar la población estudiantil destinataria, ofreciendo las herramientas adecuadas para tal fin.

Actualmente, está en desarrollo el primer prototipo de la plataforma con licencia open source. Se espera validar a FrankLab en actividades de extensión que se llevan a cabo con estudiantes de diferentes niveles educativos.

Referencias

1. P. Blikstein, A. Sipitakiat, J. Goldstein, J. Wilbert, M. Johnson, S. Vranakis, Z. Pedersen, and W. Carey. Project bloks: designing a development platform for tangible programming for children. *Position paper, retrieved online on*, pages 06–30, 2016.
2. K. Brennan and M. Resnick. New frameworks for studying and assessing the development of computational thinking. In *In AERA 2012*, 2012.
3. K.-. C. S. F. S. Committee et al. K-12 computer science framework. 2016.
4. E. Grosclaude, R. Zurita, J. Riquelme, R. del Castillo, and M. Lechner. Designing A Myro-Compatible Robot For Education As Copyleft Hardware. In *CACIC 2014*, pages 372 – 382. UNLAM, 2014.
5. E. Pasternak, R. Fenichel, and A. N. Marshall. Tips for creating a block language with blockly. In *2017 IEEE Blocks and Beyond Workshop (B B)*, pages 21–24, Oct 2017.
6. J. Rodríguez, G. Grosso, R. Zurita, and L. Cecchi. Intervención de la facultad de informática en la enseñanza de ciencias de la computación en la escuela media basada en robótica educativa. In *XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016)*, 2016.
7. T. Sapounidis and S. Demetriadis. Educational robots driven by tangible programming languages: A review on the field. In *International Conference EduRobotics 2016*, pages 205–214. Springer, 2016.
8. T. Sapounidis, I. Stamelos, and S. Demetriadis. Tangible user interfaces for programming and education: A new field for innovation and entrepreneurship. In *Innovation and Entrepreneurship in Education*, pages 271–295. Emerald Group Publishing Limited, 2016.
9. M. Smith. Computer science for all. *The White House*, 2016.
10. R. Zurita, J. d. l. Fuente, M. Bucarey, D. Bonet, R. d. Castillo, G. Grosso, L. Cecchi, J. Rodríguez, et al. Mejorando las posibilidades de aprender a programar, ampliación del robot educativo multiplo n6 max a frankestito. In *XII Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET, La Matanza 2017)*., 2017.